REMARKS/ARGUMENTS

Claims 1-18 remain pending in this application. Of these claims, claims 1, 5, 7, 11, 13, and 17 have been amended. Applicant has amended the claims to clarify the claim language. No new matter has been added to the prosecution of this application.

For at least the reasons stated below, Applicant asserts that all claims are now in condition for allowance.

OBJECTION TO SPECIFICATION

The specification was objected to for failing to support the invention's claims. Examiner further notes that the specification contains additional descriptions such that locating the portion of the disclosure that supports the claimed invention is difficult. Additionally, Examiner objected to the disclosure for using incorrect form to incorporate subject matter into this application by reference and for containing an embedded hyperlink or other form of browser-executable code.

With respect to support for the claimed invention, Examiner's attention is drawn to pages 628-632 and Figures 152-154, which clearly disclose and support each of claims 1-18. With respect to the additional descriptions in the specification, Applicant acknowledges the lengthy disclosure. Applicant respectfully submits that, by directing Examiner's attention to support for the claimed invention, that the objection should be withdrawn. If the Examiner feels this objection should stand, Applicant respectfully requests that Examiner hold this objection in abeyance until claims are allowed and Applicant has had the opportunity to amend the specification accordingly. Examiner's assistance with this matter is greatly appreciated.

With respect to the subject matter incorporated by reference, Applicant has amended the specification to include proper serial numbers and dates. Likewise, Applicant has amended the specification in accordance with MPEP § 608.01 with respect to hyperlinks or browser-executable code.

A

CLAIM REJECTIONS UNDER 35 U.S.C. § 102

Claims 1-18 are rejected under 35 U.S.C. § 102(e) as being anticipated by *Chang et al.*, U.S. patent No. 6,157,953. Applicant asserts that not every element of every claim is taught by the Chang reference. In light of the amendments and these remarks, Applicant respectfully requests the Examiner's §102 rejections be withdrawn.

MPEP § 2131 provides:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim...

Contrary to the examiner's statement that all elements are disclosed in the Chang reference, at least element (e) is not, so the rejection is unsupported by the art and should be withdrawn. The present invention provides for maintaining a security profile in nested service invocations on a distributed, component-based system, including the following elements:

(a) providing interconnections between distributed components each having nested service invocations;
(b) identifying a user;
(c) associating the user with roles;
(d) creating a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles;
(e) receiving a request from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component, and wherein completion of the second service is necessary to complete the first service;
(f) querying the user context for the information about the user;
(g) comparing the user information with an access control list for verifying that the user has access to the first component; and

A

(h) comparing the user information with an access control list for verifying that the user has access to the second service of the second component

In a typical distributed, component-based system, "a client (or user) invokes some service on a component" that in turn "may invoke any number of additional services on any number of additional components to complete its designated task." (specification, p. 629, ln. 10-13). Each successive service invocation is "a result of the initial client request so the security profile that allowed the initial request must also allow all successive requests." *Id.* The specification provides a financial example of such a system (specification, p. 629, ln. 14-26, FIG. 153):

> A user initiates an addStock() service on the Portfolio component 15300. To perform the addStock() service, the Portfolio must use the getStockPrice() and the deductFromAccount() services on the Market and Finance components 15302, 15304, respectively. This implies that a user who can access the addStock() service must also have permissions to access the getStockPrice() and the deductFromAccount() services. This may need to be checked by each of the distributed components within the context of one logical service.

Importantly, the getStockPrice and the deductFromAccount services are not merely randomly selected services; they are required in order to complete the addStock service. In other words, there is a specific relationship between the first service, addStock, and the second services, getStockPrice and deductFromAccount—namely, completion of the second services is necessary to complete the first service. This aspect of the present invention has been included in element (e) of independent claims 1, 7, and 13.

Completion of the Second Service is Necessary to Complete the First Service

Element (e) of independent claims 1, 7, and 13 requires completion of the second service in order to complete the first service. Chang describes a "method and apparatus of securing access to a service manager for the

administration of services residing on multiple service host computers..." (abstract), and is geared towards "automating the process of registering new applications and services at a central management location, such as a Web server, thereby reducing the amount of information the system administrator must remember and making a service available to end-users sooner" (col. 5, ln. 39-44).

Chang teaches services that reside on multiple host computers (see, e.g., Fig. 2) and automating login procedures to those multiple computers, thereby minimizing the number of passwords and user identifiers a user must maintain. However, no interrelationship between the various services of Chang is described. Chang merely notes that the computers are managed by a common Web server 208. Nowhere does Chang describe that "completion of the second service is necessary to complete the first service" as set forth in claims 1, 7, and 13.

Chang Does Not Describe Every Element Set Forth in Claims 1-18

As noted above, a claim is only anticipated if *every element* as set forth in the claim is found in a single prior art reference; the *identical invention* must be shown in as complete detail as is contained in the claim. For at least the reasons stated above, Chang clearly does not show the "identical invention" and "every element" of independent claims 1, 7, and 13. Accordingly, Applicant respectfully requests that the Examiner's §102 rejections as to claims 1, 7, and 13 be withdrawn.

Further, because dependent claims 2-6, 8-12, and 14-18 depend from independent claims 1, 7, and 13 respectively, Chang also fails to show every element of the dependent claims. Accordingly, Applicant respectfully requests that the Examiner's §102 rejections as to claims 2-6, 8-12, and 14-18 also be withdrawn.

Additional Arguments as to Claims 5, 11, and 17

Dependent claims 5, 11, and 17 provide for the first service associating objects with the user context, where in the object was created, updated, or deleted as a result of the invocation of the first service. (specification, p. 630, ln. 9-16; Fig. 154). Chang describes using objects for storage (col. 15, ln. 25-28) and receipt of data objects by a CPU (col. 16, ln. 2-6). However, Chang makes no other mention of objects, let alone association of objects with a user context. Nowhere does Chang describe that the "first service invoked associates any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance" as set forth in claims 5, 11, and 17.

For these additional reasons, Chang further fails to show every element of dependent claims 5, 11, and 17.

Conclusion

For at least the above-indicated reasons, Applicant submits that all pending claims are now allowable and respectfully requests that a Notice of Allowance be issued in this case. If the Examiner believes that a conference would be of value in expediting the prosecution of this application, the undersigned can be reached at the telephone number listed below.

Attached is a marked up version of the changes made to the specification by the current amendment. The attached page is captioned "Version with markings to show changes made."

A

Should any additional fees be necessary, the Commissioner is hereby authorized to charge or credit any such fees or overpayment to Deposit Account No. 50-1901 (Reference #60021-326501).

Respectfully submitted,

By _____

Steven C. Lieske, Reg. #47,749
Oppenheimer Wolff & Donnelly LLP
1400 Page Mill Road
Palo Alto, CA 94304-1124
Telephone: 612.607.7508
Facsimile: 612.607.7100
E-mail: SLieske@oppenheimer.com

VERSION WITH MARKINGS TO SHOW CHANGES MADE IN THE TITLE

Please amend the title as follows:

~~SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR~~ A USER CONTEXT COMPONENT IN ENVIRONMENT SERVICES PATTERNS.

IN THE SPECIFICATION

Please amend the specification, p. 1, ln. 7-12, as follows:

This application is related to United States Patent Applications serial number 09/387,747, filed August 31, 1999, entitled A SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR A DEVELOPMENT ARCHITECTURE FRAMEWORK, and United States Patent Applications serial number 09/387,318, filed August 31, 1999, A SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR MAINTENANCE AND ADMINISTRATION IN AN E-COMMERCE APPLICATION FRAMEWORK, both of which are filed concurrently herewith and which are incorporated by reference in their entirety.

Please amend the specification, p. 90, ln. 4-9, as follows:

The W3C also approved the specification for version 4.0 of HTML ~~(http://www.w3.org/TR/REC-html40)~~. This specification builds upon earlier iterations of HTML by enabling Web authors to include advanced forms, in-line frames, and enhanced tables in Web pages. HTML 4.0 also allows authors to publish pages in any language, and to better manage differences in language, text direction, and character encoding.

Please amend the specification, p. 92, ln. 5-15, as follows:

A number of vendors plan to use XML as the underlying language for new Web standards and applications. Microsoft uses XML for its Channel Definition Format, a Web-based "push" content delivery system included in Internet Explorer 4.0. Netscape will use XML in its Meta Content Framework

to describe and store metadata, or collections of information, in forthcoming versions of Communicator. XML is currently playing an important role the realm of electronic commerce via the Open Financial Exchange, an application developed by Microsoft, Intuit, and CheckFree for conducting electronic financial transactions. Similarly, HL7, a healthcare information systems standards organization, is using XML to support electronic data interchange EDI of clinical, financial, and administrative information ~~(http://www.mcis.duke.edu/standards/HL7/sigs/sgml/index.html)~~.

Please amend the specification, p. 92, ln. 24-p. 93, ln. 5, as follows:

In order to create 3-D worlds and objects with VRML, users need a VRML editor such as Silicon Graphics' Cosmo Worlds ~~(http://cosmo.sgi.com/products/studio/worlds)~~. To view VRML content, users need either a VRML browser or a VRML plug-in for standard HTML browsers. Leading VRML plug-ins include Cosmo Player from Silicon Graphics ~~(http://vrml.sgi.com/cosmoplayer)~~, Liquid Reality from Microsoft's DimensionX subsidiary ~~(http://www.microsoft.com/dimensionx)~~, OZ Virtual from OZ Interactive ~~(http://www.oz.com/ov/main_bot.html)~~, and WorldView from Intervista ~~(http://www.intervista.com/products/worldview/index.html)~~, These plug-ins can typically be downloaded for free from the Web.

IN THE CLAIMS

Please amend claims 1, 5, 7, 11, 13, and 17 as follows.

1. A method for maintaining a security profile throughout nested service invocations on a distributed, ~~components~~ component-based system, comprising the steps of:

(a) providing interconnections between distributed components each having nested service invocations;

(b) identifying a user;

(c) associating the user with roles;

(d) creating a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles;

(e) receiving a request from the user to invoke a first service on a first component, wherein the first component invokes a~~n additional~~ second service of ~~another~~ a second component, and wherein completion of the second service is necessary to complete the first service;

(f) querying the user context for the information about the user;

(g) comparing the user information with an access control list for verifying that the user has access to the first component; and

(h) comparing the user information with an access control list for verifying that the user has access to the ~~additional~~ second service of the ~~other~~ second component.

5. A method as recited in claim 4, wherein the first service invoked associates any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance.

7. A computer program embodied on a computer readable medium for maintaining a security profile throughout nested service invocations on a distributed, ~~components~~ component-based system, comprising:

(a) a code segment that provides interconnections between distributed components each having nested service invocations;

(b) a code segment that identifies a user;

(c) a code segment that associates the user with roles;

(d) a code segment that creates a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles;

(e) a code segment that receives a request from the user to invoke a first service on a first component, wherein the first component invokes a~~n~~

~~additional~~ second service of ~~another~~ a second component, and wherein completion of the second service is necessary to complete the first service;

(f) a code segment that queries the user context for the information about the user;

(g) a code segment that compares the user information with an access control list for verifying that the user has access to the first component; and

(h) a code segment that compares the user information with an access control list for verifying that the user has access to the ~~additional~~ second service of the ~~other~~ second component.

11. A computer program as recited in claim 10, wherein the first service invoked associates any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance.

13. A system for maintaining a security profile throughout nested service invocations on a distributed, ~~components~~ component-based system, comprising:

(a) logic that provides interconnections between distributed components each having nested service invocations;

(b) logic that identifies a user;

(c) logic that associates the user with roles;

(d) logic that creates a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles;

(e) logic that receives a request from the user to invoke a first service on a first component, wherein the first component invokes a~~n~~ ~~additional~~ second service of ~~another~~ a second component, and wherein completion of the second service is necessary to complete the first service;

(f) logic that queries the user context for the information about the user;

(g) logic that compares the user information with an access control list for verifying that the user has access to the first component; and

(h) logic that compares the user information with an access control list for verifying that the user has access to the ~~additional~~ second service of the ~~other~~ second component.

17. A system as recited in claim 16, wherein the first service invoked associates any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance.